Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach*

Ralf Küsters
University of Trier, Germany
kuesters@uni-trier.de

Tomasz Truderung
University of Trier, Germany and
Wrocław University, Poland
truderun@uni-trier.de

Abstract

In the Horn theory based approach for cryptographic protocol analysis, cryptographic protocols and (Dolev-Yao) intruders are modeled by Horn theories and security analysis boils down to solving the derivation problem for Horn theories. This approach and the tools based on this approach, including ProVerif, have been very successful in the automatic analysis of cryptographic protocols w.r.t. an unbounded number of sessions. However, dealing with the algebraic properties of operators such as the exclusive OR (XOR) has been problematic. In particular, ProVerif cannot deal with XOR.

In this paper, we show how to reduce the derivation problem for Horn theories with XOR to the XOR-free case. Our reduction works for an expressive class of Horn theories. A large class of intruder capabilities and protocols that employ the XOR operator can be modeled by these theories. Our reduction allows us to carry out protocol analysis by tools, such as ProVerif, that cannot deal with XOR, but are very efficient in the XOR-free case. We implemented our reduction and, in combination with ProVerif, applied it in the automatic analysis of several protocols that use the XOR operator. In one case, we found a new attack.

1 Introduction

In the Horn theory based approach for cryptographic protocol analysis, cryptographic protocols and the so-called Dolev-Yao intruder are modeled by Horn theories. The security analysis, including the analysis of secrecy and authentication properties, then essentially boils down to solving the derivation problem for Horn theories, i.e., the question whether a certain fact is derivable from the Horn theory. This kind of analysis takes into account that an unbounded number of protocol sessions may run concurrently. While the derivation problem is undecidable in general, there are very successful automatic analysis tools, with ProVerif [2] being one of the most promintent ones among them, which work well in practice.

However, dealing with the algebraic properties of operators, such as the exclusive OR (XOR), which are frequently used in cryptographic protocols, has been problematic in the Horn theory approach. While ProVerif has been extended to deal with certain algebraic properties in [4], associative operators, which in particular include XOR, are still out of the scope. Even though there exist some decidability results for the derivation problem in certain classes of Horn theories with XOR [9, 20, 14], the decision procedures have not led to practical implementations yet, except for the very specific setting in [14] (see the related work).

The goal of this work is therefore to come up with a practical approach that allows for the automatic analysis of a wide range of cryptographic protocols with XOR, in a setting with an unbounded number of protocol sessions. Our approach is to reduce this problem to the one without XOR, i.e., to the simpler case without algebraic properties. This simpler problem can then be solved by tools, such as ProVerif, that a priori cannot deal with XOR, but are very efficient in solving the XOR-free case. More precisely, the contribution of this paper is as follows.

Contribution of this paper. We consider an expressive class of (unary) Horn theories, called $\oplus$-linear (see Section 3). A Horn theory is $\oplus$-linear, if for every Horn clause in this theory, except for the clause that models the intruder's ability to apply the XOR operator ($I(x), I(y) \to I(x \oplus y)$), the terms that occur in these clauses are $\oplus$-linear. A term is $\oplus$-linear if for every subterm of the form $t \oplus t'$ in this term, it is true that t or t' does not contain variables. We do not put any other restriction on the Horn theories. In particular, our approach will allow us to deal with all cryptographic protocols and intruder capabilities that can be modeled as $\oplus$-linear Horn theories.

We show that the derivation problem for $\oplus$-linear Horn theories with XOR can be reduced to a purely syntactic derivation problem, i.e., a derivation problem where the al-

*An abridged version of this paper appears in CCS 2008 [16]. This work was partially supported by the DFG under Grant KU 1434/4-1, the SNF under Grant 200021-116596, and the Polish Ministry of Science and Education under Grant 3 T11C 042 30.

gebraic properties of XOR do not have to be considered anymore (see Section 3, 4, and 5). Now, the syntactic derivation problem can be solved by highly efficient tools, such as ProVerif, which cannot deal with XOR. We believe that the techniques developed in this paper are interesting beyond the case of XOR. For example, using these techniques it might be possible to also deal with other operators, such as Diffie-Hellman-Exponentiation.

Using ProVerif, we apply our two step approach—first reduce the problem, then run ProVerif on the result of the reduction—to the analysis of several cryptographic protocols that use the XOR operator in an essential way (see Section 6). The experimental results demonstrate that our approach is practical. In one case, we found a new attack on a protocol.

We note that a potential alternative to our approach is to perform unification modulo XOR instead of syntactic unification in a resolution algorithm such as the one employed by ProVerif. Whether or not this approach is practical is an open problem. The main difficulty is that unification modulo XOR is much more inefficient than syntactic unification; it is NP-complete rather than linear and, in general, there does not exist a (single) most general unifier.

Related work. In [9, 20], classes of Horn theories (security protocols) are identified for which the derivation problem modulo XOR is shown to be decidable. These classes are orthogonal to the one studied in this paper. While $\oplus$-linearity is not required, other restrictions are put on the Horn clauses, in particular linearity on the occurrence of variables. The classes in [9, 20] do, for example, not contain the Recursive Authentication and the SK3 protocol, which, however, we can model (see Section 6). To the best of our knowledge, the decision procedures proposed in [9, 20] have not been implemented. The procedure proposed in [9] has non-elementary runtime.

In [19, 14, 13], the IBM 4758 CCA API, which we also consider in our experiments, has been analyzed. Notably, in [14] a decision procedure, along with an implementation, is presented for the automatic analysis of a class of security protocols which contains the IBM 4758 CCA API. However, the protocol class and the decision procedure is especially tailored to the IBM 4758 CCA API. The only primitives that can be handled are the XOR operator and symmetric encryption. All other primitives, such as pairing, public-key encryption, and hashing, are out of the scope of the method in [14]. The specification of the IBM 4758 CCA API in [14] is hard coded in a C implementation.

In [4], it is described how the basic resolution algorithm used in ProVerif can be extended to handle some equational theories. However, as already mentioned in that work, associative operators, such as XOR, are out of the scope of this extension.

In [11], the so-called finite variant property has been studied for XOR and other operators. It has been used (implicitly or explicitly) in other works [12, 9], and also plays a role in our work (see Section 4).

In [7, 12, 15], decision procedures for protocol analysis with XOR w.r.t. a *bounded* (rather than an unbounded) number of sessions are presented. The notion of $\oplus$-linearity that we use is taken from the work in [15]. That work also contains some reduction argument. However, our work is different to [15] in several respects: First, of course, our approach is for an *unbounded* number of sessions, but it is not guaranteed to terminate. Second, the class of protocols (and intruder capabilities) we can model in our setting is much more general than the one in [15]. Third, the reduction presented in [15] heavily depends on the bounded session assumption; the argument would not work in our setting. Fourth, the reduction presented in [15] is not practical.

Structure of this paper. In Section 2, we introduce Horn theories and illustrate how they are used to model cryptographic protocols by a running example. The notion of $\oplus$-linearity is introduced in Section 3, along with a proposition that is the key to our main result, i.e., the reduction. The reduction is then presented in Section 4, with extensions to authentication presented in Section 5. We discuss our implementation and experimental results in Section 6. Proofs omitted in the main part of the paper are presented in the appendix.

We point the reader to [17] for our implementation.

2 Preliminaries

In this section, we introduce Horn theories modulo the XOR operator and illustrate how these theories are used to model the so-called Dolev-Yao intruder and cryptographic protocols by a running example.

Horn theories

Let Σ be a finite signature and V be a set of variables. The set of terms over Σ and V is defined as usual. By $\mathrm{var}(t)$ we denote the set of variables that occur in the term t. We assume Σ to contain the binary function symbol $\oplus$ (*exclusive OR*), as well as a constant 0. To model cryptographic protocols, Σ typically also contains constants (*atomic messages*), such as principal names, nonces, and keys, the unary function symbol $\mathsf{hash}(\cdot)$ (*hashing*), the unary function symbol $\mathsf{pub}(\cdot)$ (*public key*), and binary function symbols such as $\langle\cdot,\cdot\rangle$ (*pairing*), $\{\cdot\}_\cdot$ (*symmetric encryption*), and $\{\!|\cdot|\!\}_\cdot$ (*public key encryption*). The signature Σ may also contain any other free function symbol, such as various kinds of signatures and MACs. We only require that the corresponding intruder rules are $\oplus$-linear (see Section 3), which rules that do not contain the symbol $\oplus$ always are.

Ground terms, i.e. terms without variables, are called *messages*. For a unary predicate q and a (ground) term t we call $q(t)$ a *(ground) atom*. A *substitution* is a finite set of pairs of the form $\sigma = \{t_1/x_1, \ldots, t_n/x_n\}$, where $t_1, \ldots, t_n$ are terms and $x_1, \ldots, x_n$ are variables. The set $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$ is called the domain of σ. We define $\sigma(x) = x$ if $x \notin \mathrm{dom}(\sigma)$. The application $t\sigma$ of σ to a term/atom/set of terms t is defined as usual.

We call a term *standard* if its top-symbol is not $\oplus$; otherwise, it is called *non-standard*. For example, the term $\langle a, b \oplus a\rangle$ is standard, while $b \oplus a$ is non-standard.

A non-standard subterm s of t is called *complete*, if either $s = t$ or s occurs in t as a direct subterm of some standard term. For instance, for $t = \langle a \oplus \{(x \oplus y) \oplus z\}_y, b\rangle$, the terms $a \oplus \{(x \oplus y) \oplus z\}_y$ and $(x \oplus y) \oplus z$ are complete non-standard subterms of t, but $x \oplus y$ is not.

To model the algebraic properties of the exclusive OR (XOR), we consider the congruence relation $\sim$ on terms induced by the following equational theory (see, e.g., [12, 7]):

$$x \oplus y = y \oplus x \qquad (x \oplus y) \oplus z = x \oplus (y \oplus z) \qquad (1)$$

$$x \oplus x = 0 \qquad x \oplus 0 = x \qquad (2)$$

For example, we have that $t_{ex} = a \oplus b \oplus \{0\}_k \oplus b \oplus \{c \oplus c\}_k \sim a$. (Due to the associativity of $\oplus$ we often omit brackets and simply write $a \oplus b \oplus c$ instead of $(a \oplus b) \oplus c$ or $a \oplus (b \oplus c)$.) For atoms $q(t)$ and $q'(t')$, we write $q(t) \sim q'(t')$ if $q = q'$ and $t \sim t'$. We say that two terms are *equivalent modulo AC*, where AC stands for associativity and commutativity, if they are equivalent modulo (1). A term is $\oplus$*-reduced* if modulo AC, the identities (2), when interpreted as reductions from left to right, cannot be applied. Clearly, every term can be turned into $\oplus$-reduced form and this form is uniquely determined modulo AC. For example, a is the $\oplus$-reduced form of t_{ex}.

A *Horn theory* T is a finite set of *Horn clauses* of the form $a_1, \ldots, a_n \rightarrow a_0$, where a_i is an atom for every $i \in \{0, \ldots, n\}$. We assume that the variables that occur on the right-hand side of a Horn clause also occur on the left-hand side[1]. If $n = 0$, i.e., the left-hand side of the clause is always true, we call the Horn clause a_0 a *fact*.

Given a Horn theory T and a ground atom a, we say that *a can syntactically be derived from A w.r.t. T* (written $T \vdash a$) if there exists a *derivation* for a from T, i.e., there exists a sequence $\pi = b_1, \ldots, b_l$ of ground atoms such that $b_l = a$ and for every $i \in \{1, \ldots, l\}$ there exists a substitution σ and a Horn clause $a_1, \ldots, a_n \rightarrow a_0$ in T such that $a_0\sigma = b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k \in \{1, \ldots, i-1\}$ with $a_j\sigma = b_k$. In what follows, we sometimes refer to b_i by $\pi(i)$ and to $b_1, \ldots, b_i$ by $\pi_{\leq i}$. The *length l* of a derivation π is referred to by $|\pi|$.

$$\begin{array}{ll}
\mathrm{I}(x) \rightarrow \mathrm{I}(\mathsf{hash}(x)) & \mathrm{I}(x), \mathrm{I}(y) \rightarrow \mathrm{I}(\langle x, y\rangle) \\
\mathrm{I}(\langle x, y\rangle) \rightarrow \mathrm{I}(x) & \mathrm{I}(\langle x, y\rangle) \rightarrow \mathrm{I}(y) \\
\mathrm{I}(x), \mathrm{I}(y) \rightarrow \mathrm{I}(\{x\}_y), & \mathrm{I}(\{x\}_y), \mathrm{I}(y) \rightarrow \mathrm{I}(x) \\
\mathrm{I}(x), \mathrm{I}(\mathsf{pub}(y)) \rightarrow \mathrm{I}(\{\!|x|\!\}_{\mathsf{pub}(y)}), & \mathrm{I}(\{\!|x|\!\}_{\mathsf{pub}(y)}), \mathrm{I}(y) \rightarrow \mathrm{I}(x) \\
\mathrm{I}(x), \mathrm{I}(y) \rightarrow \mathrm{I}(x \oplus y) &
\end{array}$$

Figure 1: Intruder Rules.

We call a sequence $b_1, \ldots, b_l$ of ground atoms an *incomplete syntactic derivation of a from T* if $b_l = a$ and $T \cup \{b_1, \ldots, b_{i-1}\} \vdash b_i$ for every $i \in \{1, \ldots, b_l\}$.

Similarly, we write $T \vdash_\oplus a$ if there exists a *derivation of a from T modulo XOR*, i.e., there exists a sequence $b_1, \ldots, b_l$ of ground atoms such that $b_l \sim a$ and for every $i \in \{1, \ldots, l\}$ there exists a substitution σ and a Horn clause $a_1, \ldots, a_n \rightarrow a_0$ in T such that $a_0\sigma \sim b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k \in \{1, \ldots, i-1\}$ with $a_j\sigma \sim b_k$. *Incomplete derivations modulo XOR* are defined analogously to the syntactic case.

Given T and a, we call the problem of deciding whether $T \vdash a$ ($T \vdash_\oplus a$) is true, the *deduction problem (modulo XOR)*. In case T models a protocol and the intruder (as described below), the fact that $T \vdash_\oplus a$, with $a = \mathrm{I}(t)$, is *not* true means that the term t is secret, i.e., the intruder cannot get hold of t even when running an unbounded number of sessions of the protocol and using algebraic properties of the XOR operator.

Modeling Protocols by Horn theories

Following [2], we now illustrate how Horn theories can be used to analyze cryptographic protocols, where, however, we take the XOR operator into account. While here we concentrate on secrecy properties, authentication is discussed in Section 5. As mentioned in the introduction, the Horn theory approach allows us to analyze the security of protocols w.r.t. an unbounded number of sessions and with no bound on the message size in a fully automatic and sound way. However, the algorithms are not guaranteed to terminate and may produce false attacks.

A Horn theory for modeling protocols and the (Dolev-Yao) intruder uses only the predicate I. The fact $\mathrm{I}(t)$ means that the intruder may be able to obtain the term t. The fundamental property is that if $\mathrm{I}(t)$ cannot be derived from the set of clauses, then the protocol preserves the secrecy of t. The Horn theory consists of three sets of Horn clauses: the initial intruder facts, the intruder rules, and the protocol rules. The set of *initial intruder facts* represents the

[1] This assumption can easily be relaxed for variables that are substituted only be cetrain "good" terms, where "good" means C-dominated (see Section 3)

initial intruder knowledge, such as names of principals and public keys. The clauses in this set are facts, e.g., $\mathrm{I}(a)$ (the intruder knows the name a) and $\mathrm{I}(\mathsf{pub}(sk_a))$ (the intruder knows the public key of a, with sk_a being the corresponding private key). The set of *intruder rules* represents the intruders ability to derive new messages. For the cryptographic primitives mentioned above, the set of intruder rules consists of the clauses depicted in Figure 1. The last clause in this figure will be called the $\oplus$*-rule*. It allows the intruder to perform the XOR operation on arbitrary messages. The set of *protocol rules* represents the actions performed in the actual protocol. The ith protocol step of a principal is described by a clause of the form $\mathrm{I}(r_1), \dots, \mathrm{I}(r_i) \to \mathrm{I}(s_i)$ where the terms r_j, $j \in \{1, \dots, i\}$, describe the (patterns of) messages the principal has received in the previous $i-1$st steps plus the (pattern of the) message in the ith step. The term $\mathrm{I}(s_i)$ is the (pattern of) the ith output message of the principal. Given a protocol P, we denote by T_P the Horn theory that comprises all three sets mentioned above.

Let us illustrate the above by a simple example protocol, which we will use as a running example throughout this paper. Applications of our approach to more complex protocols are presented in Section 6.2. We emphasize that the kind of Horn theories outlined above are only an example of how protocols and intruders can be modeled. As already mentioned in the introduction, our methods applies to all $\oplus$-linear Horn theories.

Running example

We consider a protocol that was proposed in [7]. It is a variant of the Needham-Schroeder-Lowe protocol in which XOR is employed. The informal description of the protocol, which we denote by $P_{NSL_\oplus}$, is as follows:

$$\begin{array}{lll} (1) & A \to B: & \{\!|\langle N, A\rangle|\!\}_{\mathsf{pub}(sk_B)} \\ (2) & B \to A: & \{\!|\langle M, N \oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)} \\ (3) & A \to B: & \{\!|M|\!\}_{\mathsf{pub}(sk_B)} \end{array}$$

where N and M are nonces generated by A and B, respectively. As noted in [7], this protocol is insecure; a similar attack as the one on the original Needham-Schroeder protocol can be mounted, where, however, now the algebraic properties of XOR are exploited.

To illustrate how this protocol can be modeled in terms of Horn theories, let P be a set of participant names and $\mathsf{H} \subseteq \mathsf{P}$ be the set of names of the honest participants. As proved in [10], for the secrecy property it suffices to consider the case $\mathsf{P} = \{a, b\}$ and $\mathsf{H} = \{a\}$ (for authentication three participants are needed). In the following, sk_a, for $a \in \mathsf{P}$, denotes the private key of a, $n(a, b)$ denotes the nonce sent by $a \in \mathsf{P}$ to $b \in \mathsf{P}$ in message 1., and $m(b, a)$ denotes the nonce generated by b and sent to a in message 2.

The initial intruder knowledge is the set $\{\mathrm{I}(a) \mid a \in \mathsf{P}\} \cup \{\mathrm{I}(\mathsf{pub}(sk_a)) \mid a \in \mathsf{P}\} \cup \{\mathrm{I}(sk_a) \mid a \in \mathsf{P} \setminus \mathsf{H}\}$ of facts. The intruder rules are those depicted in Figure 1. The first step of the protocol performed by an honest principal is modeled by the facts:

$$\mathrm{I}(\{\!|\langle n(a,b), a\rangle|\!\}_{\mathsf{pub}(sk_b)})$$

for $a \in \mathsf{H}$, $b \in \mathsf{P}$. Note that it is not necessary to model messages sent by dishonest principals, since these are taken care of by the actions that can be performed by the intruder.

The second step of the protocol performed by an honest principal is modeled by the clauses:

$$\mathrm{I}(\{\!|\langle x, a\rangle|\!\}_{\mathsf{pub}(sk_b)}) \to \mathrm{I}(\{\!|\langle m(b,a), x \oplus b\rangle|\!\}_{\mathsf{pub}(sk_a)}) \qquad (3)$$

for $b \in \mathsf{H}$, $a \in \mathsf{P}$. The third step of the protocol performed by an honest principal is modeled by the clauses:

$$\mathrm{I}(\{\!|\langle y, n(a,b) \oplus b\rangle|\!\}_{\mathsf{pub}(sk_a)}) \to \mathrm{I}(\{\!|y|\!\}_{\mathsf{pub}(sk_b)}) \qquad (4)$$

for $a \in \mathsf{H}$, $b \in \mathsf{P}$. The set of Horn clauses defined above is denoted by $T_{P_{NSL_\oplus}}$. It is not hard to verify that we have $T_{P_{NSL_\oplus}} \vdash_\oplus m(b, a)$ for every $a, b \in \mathsf{H}$. In fact, secrecy of the nonces sent by an honest responder to an honest initiator is not guaranteed by the protocol [7].

3 Dominated Derivations

In Section 4, we show how to reduce the deduction problem modulo XOR to the one without XOR for $\oplus$-linear Horn theories, introduced below. This reduction allows us to reduce the problem of checking secrecy for protocols that use XOR to the case of protocols that do not use XOR. (The authentication problem will be considered in Section 5.) The latter problem can then be solved by tools that cannot deal with XOR, such as ProVerif. The class of protocol and intruder capabilities that we can handle this way is quite large: It contains all protocol and intruder rules that are $\oplus$-linear.

In this section, we prove a proposition that will be the key to the reduction. Before we can state the proposition, we need to introduce $\oplus$-linear Horn theories and some further terminology.

A term is $\oplus$*-linear* if for each of its subterms of the form $t \oplus s$, where t and s may be standard or non-standard terms, it is true that t or s is ground. In other words, if a term t contains a subterm of the form $t_1 \oplus \cdots \oplus t_n$ with $n \geq 2$, t_i standard for every i, and there exists i and j, $i \neq j$, such that t_i and t_j are not ground, then t is not $\oplus$-linear. For example, for variables x, y, z and a constant a, the term $t^1_{ex} = \langle a, a \oplus \langle x, y\rangle\rangle$ is $\oplus$-linear, but the term $t^2_{ex} = \langle a, a \oplus \langle x, y\rangle \oplus z\rangle$ is not. A Horn clause is called $\oplus$-linear if each term occurring in the clause is $\oplus$-linear. A Horn theory is $\oplus$-linear if each clause in this theory, except for the $\oplus$-rule (see Fig. 1), is $\oplus$-linear. In particular, given a protocol P, the induced theory T_P is $\oplus$-linear if the sets of protocol and intruder rules, except for the $\oplus$-rule, are.

Our running example is an example of a protocol with an $\oplus$-linear Horn theory (note that, in (3) and (4), b is a constant); other examples are mentioned in Section 6.2. Also, many intruder rules are $\oplus$-linear. In particular, all those that do not contain the XOR symbol. For example, in addition to the cryptographic primitives mentioned in Figure 1, other primitives, such as various kinds of signatures, encryption with prefix properties, and MACs have $\oplus$-linear intruder rules.

Besides $\oplus$-linearity, we also need a more fine-grained notion: C-domination. Let C be a finite set of standard $\oplus$-reduced ground terms such that C does not contain two elements m, m' with $m \neq m'$ and $m \sim m'$. (For the efficiency of our reduction (Section 4), it is important to keep C as small as possible.) Let $\mathsf{C}^{\oplus} = \{t \mid \text{there exist } c_1, \ldots, c_n \in \mathsf{C} \text{ such that } t \sim c_1 \oplus \cdots \oplus c_n\}$ be the $\oplus$-closure of C. Note that $0 \in \mathsf{C}^{\oplus}$. Finally, let $\tilde{\mathsf{C}} = \{t \mid t \sim t' \in \mathsf{C}, t \text{ standard}\}$.

Now, a term is C-*dominated* if, for each of its subterms of the form $t \oplus s$, where t and s may be standard or non-standard, it is true that t or s is in $\mathsf{C}^{\oplus}$. For example, the term t^1_{ex} from above is $\{a\}$-dominated, but is is not $\{b\}$-dominated. The term t^2_{ex} is not $\{a\}$-dominated. A Horn clause is C-dominated, if the terms occurring in this clause are C-dominated; similarly for derivations. Finally, a Horn theory T is C-dominated if each clause in T, except for the $\oplus$-rule, is C-dominated. For example, we have that the Horn theory $T_{P_{NSL_{\oplus}}}$ of our running example is $\{a, b\}$-dominated. (Recall that $\mathsf{P} = \{a, b\}$.)

C-dominated terms can also be characterized in terms of what we call bad terms. We call a non-standard term t *bad* (w.r.t. C), if $t \sim c \oplus t_1 \oplus \ldots \oplus t_n$ for $c \in \mathsf{C}^{\oplus}$, pairwise $\oplus$-distinct standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, and $n > 1$, where t and t' are $\oplus$-*distinct* if $t \not\sim t'$. A non-standard term which is not bad is called *good*. The following lemma is easy to see:

Lemma 1. *An $\oplus$-reduced term is C-dominated iff it contains no bad subterms.*

There is an obvious connection between $\oplus$-linearity and C-domination:

Lemma 2. *For every $\oplus$-linear term/Horn theory/derivation there exists a finite set C of standard $\oplus$-reduced messages such that the term/Horn theory/derivation is C-dominated.*

The set C mentioned in the lemma could be chosen to be the set of all ground standard terms occurring in the term/Horn theory/derivation. However, C should be chosen as small as possible in order to make the reduction presented in Section 4 more efficient.

As mentioned, the following proposition is the key to our reduction. The proposition states that C-dominated Horn theories always allow for C-dominated derivations. Because of Lemma 2, the proposition applies to all $\oplus$-linear Horn theories.

Proposition 1. *Let T be a C-dominated Horn theory and b be a C-dominated fact. If $T \vdash_{\oplus} b$, then there exists a C-dominated derivation modulo XOR for b from T.*

Before we present the proof of this proposition, we introduce some terminology, which is also used in subsequent sections, and sketch the idea of the proof. We write $t \simeq_{\mathsf{C}} t'$ if $t' \sim c \oplus t$ (or equivalently, $c \oplus t' \sim t$), for some $c \in \mathsf{C}^{\oplus}$.

For the rest of this section we fix a derivation π modulo XOR for b from T. W.l.o.g. we may assume that each term occurring in π is in $\oplus$-reduced form and that each term in a substitution applied in π is in $\oplus$-reduced form as well.

The key definitions for the proof of Proposition 1 are the following ones:

Definition 1. For a standard term t, the set C, and the derivation π, we define the *type of t (w.r.t. π and C)*, written $\tilde{t}$, to be an $\oplus$-reduced element c of $\mathsf{C}^{\oplus}$ such that $\pi(i) \sim \mathrm{I}(c \oplus t)$ for some i, and for each $j < i$, it is not true that $\pi(j) \sim \mathrm{I}(c' \oplus t)$ for some $c' \in \mathsf{C}^{\oplus}$. If such an i does not exist, we say that the type of t is undefined.

Note that the type of a term is uniquely determined modulo AC and that equivalent terms (w.r.t. $\sim$) have equivalent types.

In the following definition, we define an operator which replaces standard terms in bad terms which are not in $\tilde{\mathsf{C}}$ by their types. This turns a bad term into a good one. To define the operator, we use the following notation. We write $\varphi_{\oplus}[x_1, \ldots, x_n]$ for a term which is built only from $\oplus$, elements of $\tilde{\mathsf{C}}$, and the pairwise distinct variables $x_1, \ldots, x_n$ such that each x_i occurs exactly once in $\varphi_{\oplus}[x_1, \ldots, x_n]$. An example is $\varphi^{ex}_{\oplus}[x_1, x_2, x_3] = ((x_1 \oplus x_2) \oplus (a \oplus x_3))$, where $a \in \tilde{\mathsf{C}}$. For messages $t_1, \ldots, t_n$, we write $\varphi_{\oplus}[t_1, \ldots, t_n]$ for the message obtained from $\varphi_{\oplus}[x_1, \ldots, x_n]$ by replacing every x_i by t_i, for every $i \in \{1, \ldots, n\}$. Note that each non-standard term can be expressed in the form $\varphi_{\oplus}[t_1, \ldots, t_n]$ for some $\varphi_{\oplus}$ as above and standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$.

Definition 2. For a message t, we define $\Delta(t)$ as follows: If t is a bad term of the form $\varphi_{\oplus}[t_1, \ldots, t_n]$ for some $\varphi_{\oplus}$ as above and standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, then $\Delta(t) = \varphi_{\oplus}[\tilde{t}_1, \ldots, \tilde{t}_n]$; $\Delta(t)$ is undefined, if one of those $\tilde{t}_i$ is undefined. Otherwise (if t is good), we recursively apply Δ to all direct subterms of t.

We will see (Lemma 10) that if t occurs in π, then the types of t_i in the above definition are always defined. Note also that Δ is defined with respect to the given π and C.

Now, the main idea behind the proof of Proposition 1 is to apply $\Delta(\cdot)$ to π. We then show that (i) $\Delta(\pi)$ is an incomplete C-dominated derivation modulo XOR for b from T and (ii) to obtain a complete derivation only C-dominated terms are needed. The details of the proof are presented next, by a series of lemmas, some of which are also used in Section 4.

Proof of Proposition 1. The following lemma is easy to show by structural induction on s:

Lemma 3. *Let s and t be messages such that s is $\oplus$-reduced, s contains a complete bad subterm s', and $s \sim t$. Then, there exists a complete bad subterm t' of t such that $t' \sim s'$.*

The following lemma, whose proof can be found in the appendix, says that when substituting variables in a C-dominated term, then complete bad terms that might have been introduced by the substitution cannot be canceled out by the C-dominated term.

Lemma 4. *Let $r\theta \sim t$, for a term t, an $\oplus$-reduced substitution θ, and a C-dominated term r. Then, for each complete bad subterm r' of $r\theta$ there exists a complete (bad) subterm t' of t such that $t' \sim r'$.*

We now show (see the appendix) that if an instance of a C-dominated term contains a complete bad subterm, then this term (up to $\simeq_{\mathsf{C}}$) must be part of the substitution with which the instance was obtained.

Lemma 5. *Let θ be a ground substitution and s be a C-dominated term. Assume that t is a complete bad subterm of $s\theta$. Then, there exists a variable x and a complete bad subterm t' of $\theta(x)$ such that $t' \simeq_{\mathsf{C}} t$.*

The converse of Lemma 5 is also easy to show by structural induction on s.

Lemma 6. *Let θ be a ground substitution and s be a C-dominated term. If $s\theta$ is C-dominated, then so is $\theta(x)$ for every $x \in \mathrm{var}(s)$.*

Similarly to Lemma 5, we can prove the following lemma. The main observation is that $\Delta(c \oplus t) \sim c \oplus \Delta(t)$, for $c \in \mathsf{C}^{\oplus}$.

Lemma 7. *$\Delta(s\theta) \sim s(\Delta\theta)$, for a C-dominated term s and a substitution θ.*

Another basic and simple to prove property of Δ is captured in the following lemma.

Lemma 8. *Let s and t be terms such that $s \sim t$. Then, $\Delta(s) \sim \Delta(t)$.*

The following lemma says that if an instance of a C-dominated Horn clause contains a complete bad subterm on its right-hand side, then this term (up to $\simeq_{\mathsf{C}}$) already occurs on the left-hand side.

Lemma 9. *Assume that $p_1(r_1), \ldots, p_n(r_n) \rightarrow p_0(s)$ is a C-dominated Horn clause, θ is an $\oplus$-reduced ground substitution, $w, u_1, \ldots, u_n$ are $\oplus$-reduced messages such that $w \sim s\theta$ and $u_i \sim r_i\theta$, for $i \in \{1, \ldots, n\}$.*

If w' is a complete bad subterm of w, then there exists a complete bad subterm u' of u_i, for some $i \in \{1, \ldots, n\}$, such that $u' \simeq_{\mathsf{C}} w'$.

Proof. Suppose that w' is a complete bad subterm of w. Because $w \sim s\theta$ and w is $\oplus$-reduced, by Lemma 3, there exists a complete bad subterm t of $s\theta$ with $w' \sim t$. By Lemma 5, there exists a variable $x \in \mathrm{var}(s)$ and a complete bad subterm t' of $\theta(x)$ with $t' \simeq_{\mathsf{C}} t$. Because x, as a variable of s, has to occur also in r_i for some $i \in \{1, \ldots, n\}$, the term t' is a (not necessarily complete) subterm of $r_i\theta$. Since r_i is C-dominated, there exists a complete subterm r' of $r_i\theta$ with $r' \simeq_{\mathsf{C}} t'$. Now, recall that $t' \simeq_{\mathsf{C}} t$ and $t \sim w'$. It follows that $r' \simeq_{\mathsf{C}} w'$. Furthermore, since w' is bad, so is r'. Now, by Lemma 4, there exists a complete bad subterm u' of u_i such that $u' \simeq_{\mathsf{C}} r' \simeq_{\mathsf{C}} w'$. □

The following lemma connects bad terms that occur in a derivation with the types of their subterms.

Lemma 10. *For every $n \geq 1$, if $\pi(i) \sim \mathrm{I}(c \oplus t_1 \oplus \cdots \oplus t_n)$, for $c \in \mathsf{C}^{\oplus}$ and pairwise $\oplus$-distinct standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, then, for each $k \in \{1, \ldots, n\}$, there exists $j \leq i$ such that $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$.*

Proof. If $n = 1$, then $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ belongs to $\pi_{\leq i}$, by the definition of types.

Now, suppose that $n > 1$. In that case we will show, by induction on i, something more than what is claimed in the lemma: If t with $t \sim c \oplus t_1 \oplus \cdots \oplus t_n$, $c \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct standard terms $t_i \notin \tilde{\mathsf{C}}$, occurs as a complete bad subterm in $\pi(i)$, then, for each $k \in \{1, \ldots, n\}$, there exists $j \leq i$ such that $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$.

Suppose that t, as above, occurs as a complete bad subterm in $\pi(i)$.

If there exists t' such that $t' \simeq_{\mathsf{C}} t$ and t' occurs in $\pi_{<i}$ as a complete subterm, then we are trivially done by the induction hypothesis. (Note that t' is bad since t is.) So, suppose that such a t' does not occur in $\pi_{<i}$ as a complete subterm. By Lemma 9, $\pi(i)$ cannot be obtained by a C-dominated Horn clause. Thus, $\pi(i)$ is obtained by the $\oplus$-rule, which means that $\pi(i) = \mathrm{I}(u)$ with $u \sim s \oplus r$ for some $\mathrm{I}(s)$ and $\mathrm{I}(r)$ occurring in $\pi_{<i}$. We may assume that $s \sim d \oplus s_1 \oplus \cdots \oplus s_p$, with $d \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct $\oplus$-reduced standard terms $s_1, \ldots, s_p \notin \tilde{\mathsf{C}}$, and $r \sim e \oplus r_1 \oplus \cdots \oplus r_q$, with $e \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct $\oplus$-reduced standard terms $r_1, \ldots, r_q \notin \tilde{\mathsf{C}}$.

According to our assumption, neither s nor r contains a complete subterm t' with $t' \simeq_{\mathsf{C}} t$. In particular, neither s nor r contains t' with $t' \sim t$. So, since $\pi(i) \sim \mathrm{I}(s \oplus r)$ contains t as a complete subterm, it must be the case that

$t \sim s \oplus r$. Now, with $t \sim c \oplus t_1 \oplus \ldots \oplus t_n$, as above, and $k \in \{1,\ldots,n\}$ it follows that either $s_l \sim t_k$ or $r_l \sim t_k$, for some l. Suppose that the former case holds (the argument is similar for the latter case). If $p > 1$ (and thus s is a bad term), then, by the induction hypothesis, we know that there exists $j < i$ such that $\pi(j) \sim \mathrm{I}(\tilde{s}_l \oplus s_l)$. Since $t_k \sim s_l$, we have that $\tilde{t}_k \sim \tilde{s}_l$, and hence, $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$. Otherwise, $s \sim d \oplus t_k$, and hence, by the definition of types, there exists $j < i$ with $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$. □

The following lemma is the key in proving that $\Delta(\pi)$ is an incomplete derivation modulo XOR.

Lemma 11. *For every $i \leq |\pi|$, if $\mathrm{I}(c \oplus t_1 \oplus \cdots \oplus t_n)$, for some $c \in \mathsf{C}^{\oplus}$ and pairwise $\oplus$-distinct standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, belongs to $\pi_{<i}$, then there is a derivation for $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n)$ from $\Delta(\pi_{<i})$ modulo XOR.*

Proof. If $n = 0$ or $n > 1$, then $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n) \sim \mathrm{I}(\Delta(c \oplus t_1 \oplus \cdots \oplus t_n))$ by the definition of Δ, and hence, $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n)$ can be derived from $\Delta(\pi_{<i})$. So suppose that $n = 1$. Since we have $\mathrm{I}(c \oplus t_1)$ in $\pi_{<i}$, then, by the definition of types, we also have $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ in $\pi_{<i}$. Thus, by the definition of Δ, $\mathrm{I}(c \oplus \Delta(t_1))$ and $\mathrm{I}(\tilde{t}_1 \oplus \Delta(t_1))$ are in $\Delta(\pi_{<i})$. From these one obtains $\mathrm{I}(c \oplus \tilde{t}_1)$ by applying the $\oplus$-rule. □

Now, we can finish the proof of Proposition 1. First, note that every non-standard message in $\Delta(\pi)$ is C-dominated. This immediately follows from the definition of Δ. We will now show (*): For each $i \in \{1, \ldots, |\pi|\}$, $\Delta(\pi(i))$ can be derived from $\Delta(\pi_{<i})$ modulo XOR by using only C-dominated terms. This then completes the proof of Proposition 1.

Recall that we assume that π is $\oplus$-reduced and that in this derivation we use only $\oplus$-reduced substitutions. To prove (*), we consider two cases:

Case 1. $\pi(i)$ is obtained from $\pi_{<i}$ using a C-dominated Horn clause $R = (p_1(s_1), \ldots, p_n(s_n) \rightarrow p_0(s_0))$ of T: Then there exists a $\oplus$-reduced substitution θ such that $\pi(i) \sim p_0(s_0\theta)$ and the atoms $p_1(s_1\theta), \ldots, p_n(s_n\theta)$ occur in $\pi_{<i}$ modulo XOR. Thus, by Lemma 8, $p_1(\Delta(s_1\theta)), \ldots, p_n(\Delta(s_n\theta))$ occur in $\Delta(\pi_{<i})$ modulo XOR. Now, by Lemma 7, we have that $\Delta(s_i\theta) \sim s_i(\Delta\theta)$, for every $i \in \{0, \ldots, n\}$. Thus, by applying R with the substitution $\Delta(\theta)$, we obtain $\Delta(\pi(i)) \sim \Delta(s_0\theta) \sim s_0(\Delta(\theta))$.

Case 2. $\pi(i)$ is obtained by the $\oplus$-rule: Hence, there are two atoms $\mathrm{I}(s)$ and $\mathrm{I}(r)$ in $\pi_{<i}$ such that $\pi(i) \sim \mathrm{I}(s \oplus r)$. We may assume that $s \sim c \oplus s_1 \oplus \cdots \oplus s_m$, with $c \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct $\oplus$-reduced standard terms $s_1, \ldots, s_m \notin \tilde{\mathsf{C}}$, and $r \sim d \oplus r_1 \oplus \cdots \oplus r_l$, with $d \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct $\oplus$-reduced standard terms $r_1, \ldots, r_l \notin \tilde{\mathsf{C}}$. Let $\{t_1, \ldots, t_n\} = (S \setminus R) \cup (R \setminus S)$, for $S = \{s_1, \ldots, s_m\}$ and $R = \{r_1, \ldots, r_l\}$. Then, $\pi(i) \sim \mathrm{I}(s \oplus r) \sim I(c \oplus d \oplus t_1 \oplus \cdots \oplus t_n)$. By Lemma 11, we know that $\mathrm{I}(c \oplus \tilde{s}_1 \oplus \cdots \oplus \tilde{s}_m)$ and $\mathrm{I}(d \oplus \tilde{r}_1 \oplus \cdots \oplus \tilde{r}_l)$ can be derived from $\Delta(\pi_{<i})$ modulo XOR. Hence, $\mathrm{I}(t')$ with $t' = c \oplus d \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n$ can be derived from $\Delta(\pi_{<i})$ as well (by applying the $\oplus$-rule). Now, let us consider two cases:

(a) $n = 0$ or $n > 1$: In this case, we have that $\Delta(\pi(i)) \sim \mathrm{I}(t')$, and hence, $\Delta(\pi(i))$ can be derived from $\Delta(\pi_{<i})$.

(b) $n = 1$: Because $\mathrm{I}(c \oplus s_1 \oplus \cdots \oplus s_m)$ and $\mathrm{I}(d \oplus r_1 \oplus \cdots \oplus r_l)$ occur in $\pi_{<i}$ modulo XOR, by Lemma 10, $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ occurs in $\pi_{<i}$ modulo XOR as well. Thus, by Lemma 8, $\mathrm{I}(\tilde{t}_1 \oplus \Delta(t_1))$ occurs in $\Delta(\pi_{<i})$ modulo XOR. Now, because $\mathrm{I}(t')$, with $t' = c \oplus d \oplus \tilde{t}_1$, can be derived from $\Delta(\pi_{<i})$ modulo XOR, so can $\mathrm{I}(c \oplus d \oplus \Delta(t_1)) \sim \Delta(\pi(i))$. □

4 The Reduction

In this section, we show how the deduction problem modulo XOR can be reduced to the deduction problem without XOR for C-dominated theories. More precisely, for a C-dominated theory T, we show how to effectively construct a Horn theory T^+ such that a (C-dominated) fact can be derived from T modulo XOR iff it can be derived from T^+ in a syntactic derivation, where XOR is considered to be a function symbol without any algebraic properties. As mentioned, the syntactic deduction problem, and hence, the problem of checking secrecy for cryptographic protocols w.r.t. an unbounded number of sessions, can then be solved by tools, such as ProVerif, which cannot deal with the algebraic properties of XOR.

In the remainder of this section, let T be a C-dominated theory. In what follows, we will first define the reduction function, which turns T into T^+, and state the main result (Section 4.1), namely that the reduction is sound and complete as stated above. Before proving this result in Section 4.3, we illustrate the reduction function by our running example (Section 4.2).

4.1 The Reduction Function

The reduction function uses an operator $\ulcorner \cdot \urcorner$, which turns terms into what we call normal form, and a set $\Sigma(t)$ of substitutions associated with the term t. We first define this operator and the set $\Sigma(t)$. The operator $\ulcorner \cdot \urcorner$ is defined w.r.t. a linear ordering $<_{\mathsf{c}}$ on C, which we fix once and for all.

Definition 3. For a C-dominated term t, we define the *normal form* of t, denoted by $\ulcorner t \urcorner$, recursively as follows:

- If t is a variable, then $\ulcorner t \urcorner = t$.
- If $t = f(t_1, \ldots, t_n)$ is standard, then $\ulcorner t \urcorner = f(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner)$.
- If $t \in \mathsf{C}^{\oplus}$ is non-standard and $t \sim c_1 \oplus \cdots \oplus c_n$, for some pairwise $\oplus$-distinct $c_1, \ldots, c_n \in \mathsf{C}$, $n > 1$, such

that $c_1 <_{\mathsf{C}} \cdots <_{\mathsf{C}} c_n$, then $\ulcorner t \urcorner = \ulcorner c_1 \urcorner \oplus (\ulcorner c_2 \urcorner \oplus (\cdots \oplus \ulcorner c_n \urcorner) \cdots)$.

- If t is non-standard and $t \sim c \oplus t'$, for some $c \in \mathsf{C}^{\oplus}$, $c \not\sim 0$, and standard t' not in $\tilde{\mathsf{C}}$, then $\ulcorner t \urcorner = \ulcorner c \urcorner \oplus \ulcorner t' \urcorner$.

We say that a term t is in *normal form*, if $t = \ulcorner t \urcorner$. A substitution θ is in normal form, if $\theta(x)$ is in normal form for each variable x in the domain of θ.

It is easy to see that $\ulcorner t \urcorner = \ulcorner s \urcorner$ for C-dominated terms t and s iff $t \sim s$, and that $\ulcorner t \urcorner$ is $\oplus$-reduced for any t. By $\mathsf{C}^{\oplus}_{\mathsf{norm}}$, we denote the set $\{\ulcorner c \urcorner \mid c \in \mathsf{C}^{\oplus}\}$. Clearly, this set is finite and computable in exponential time in the size of C.

To define the set $\Sigma(t)$ of substitutions, we need the notion of fragile subterms. For a C-dominated term t, the set of *fragile subterms of* t, denoted by $\mathcal{F}(t)$, is $\mathcal{F}(t) = \{s \mid s$ is a non-ground, standard term which occurs as a subterm of t in the form $t' \oplus s$ or $s \oplus t'$ for some $t'\}$. For example, $\mathcal{F}((a \oplus \langle x, b\rangle) \oplus b) = \{\langle x, b\rangle\}$.

We are now ready to define the (finite and effectively computable) set $\Sigma(t)$ of substitutions for a C-dominated term t. The main property of this set is the following: For every C-dominated, ground substitution θ in normal form, there exists a substitution $\sigma \in \Sigma(t)$ and a substitution θ' such that $\ulcorner t\theta \urcorner = (\ulcorner t\sigma \urcorner)\theta'$. In other words, the substitutions in $\Sigma(t)$ yield all relevant instances of t. All ground, normalized instances are syntactic instances of those instances. This resembles the finite variant property of XOR [11] mentioned in the introduction. However, our construction of $\Sigma(t)$ is tailored and optimized towards C-dominated terms and substitutions. More importantly, we obtain a stronger property in the sense that the equality—$\ulcorner t\theta \urcorner = (\ulcorner t\sigma \urcorner)\theta'$—is *syntactic* equality, not only equality modulo AC; the notion of C-domination, which we introduced here, is crucial in order to obtain this property. Having syntactic equality is important for our reduction in order to get rid of algebraic properties completely.

Definition 4. Let t be a C-dominated term. We define a family of substitutions $\Sigma(t)$ as follows. The domain of every substitution in $\Sigma(t)$ is the set of all variables which occur in some $s \in \mathcal{F}(t)$. Now, $\sigma \in \Sigma$, if for each $x \in \mathrm{dom}(\sigma)$ one of the following cases holds:

(i) $\sigma(x) = x$,

(ii) $x \in \mathcal{F}(t)$ and $\sigma(x) = c \oplus x$, for some $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$, $c \neq 0$,

(iii) there exists $s \in \mathcal{F}(t)$ with $x \in \mathrm{var}(s)$ and a C-dominated substitution θ in normal form such that $s\theta \in \mathsf{C}^{\oplus}$ and $\sigma(x) = \theta(x)$.

To illustrate the definition and the property mentioned above, consider, as an example, $t = c \oplus x$ and the substitution $\theta(x) = d \oplus m$, with $d \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and a C-dominated, standard term $m \notin \mathsf{C}^{\oplus}_{\mathsf{norm}}$ in normal form. In this case, we can choose $\sigma(x) = d \oplus x$ according to (ii). With $\theta'(x) = m$, we obtain $\ulcorner t\theta \urcorner = \ulcorner c \oplus d \urcorner \oplus m = (\ulcorner t\sigma \urcorner)\theta'$. If $\theta(x)$ were $d \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$, then (iii) would be applied.

We can show (see the appendix):

Lemma 12. *For a C-dominated term t, the set $\Sigma(t)$ can be computed in exponential time in the size of t.*

We are now ready to define the reduction function which turns T into T^+. The Horn theory T^+ is given in Fig. 2. With the results shown above, it is clear that T^+ can be constructed in exponential time from T. The Horn clauses in (6)–(9) simulate the $\oplus$-rule in case the terms we consider are C-dominated. The other rules in T are simulated by the rules in (5), which are constructed in such a way that they allow us to produce messages in normal form for input messages in normal form.

We can now state the main theorem of this paper. This theorem states that a message (a secret) can be derived from T using derivations modulo XOR if and only if it can be derived from T^+ using only syntactic derivations, i.e., no algebraic properties of XOR are taken into account. As mentioned, this allows to reduce the problem of verifying secrecy for cryptographic protocols with XOR, to the XOR-free case. The latter problem can then be handled by tools, such as ProVerif, which otherwise could not deal with XOR.

Theorem 1. *For a C-dominated Horn theory T and C-dominated message b in normal form, we have: $T \vdash_{\oplus} b$ if and only if $T^+ \vdash b$.*

Before we prove this theorem, we illustrate the reduction by our running example.

4.2 Example

Consider the Horn theory $T_{P_{NSL_{\oplus}}}$ of our running example. As mentioned in Section 3, this Horn theory is C-dominated for $\mathsf{C} = \{a, b\}$. In what follows, we illustrate how $T^+_{P_{NSL_{\oplus}}}$ looks like, where the elements of C are ordered as $a <_{\mathsf{C}} b$.

First, consider the instances of Horn clauses of $T_{P_{NSL_{\oplus}}}$ given by (5). Only the Horn clauses in (3) have fragile subterms. All other Horn clauses have only one instance in $T^+_{P_{NSL_{\oplus}}}$: the rule itself. This is because for such Horn clauses $\Sigma(\cdot)$ contains only one substitution, the identity. The Horn clause in (3) has one fragile subterm, namely x. Hence, the domain of every substitution in the corresponding Σ-set is $\{x\}$, and according to Definition 4, this set contains the following eight substitutions: item (i) gives $\sigma_1 = \{x/x\}$; item (ii) gives $\sigma_2 = \{a \oplus x/x\}$, $\sigma_3 = \{b \oplus x/x\}$, and $\sigma_4 = \{(a \oplus b) \oplus x/x\}$; item (iii) gives $\sigma_5 = \{0/x\}$, $\sigma_6 = \{a/x\}$, $\sigma_7 = \{b/x\}$, and $\sigma_8 = \{a \oplus b/x\}$. For each of these substitutions we obtain an instance of (3). For example, σ_4 yields

$$\mathrm{I}(\{\!\!\{\langle (a \oplus b) \oplus x, a\rangle\}\!\!\}_{\mathsf{pub}(sk_b)}) \to \mathrm{I}(\{\!\!\{\langle m(b, a), a \oplus x\rangle\}\!\!\}_{\mathsf{pub}(sk_a)}).$$

$$\ulcorner r_1\sigma\urcorner,\dots,\ulcorner r_n\sigma\urcorner \to \ulcorner r_0\sigma\urcorner \qquad \text{for each } \mathsf{C}\text{-dominated rule } r_1,\dots,r_n \to r_0 \text{ of } T \text{ and each } \sigma \in \Sigma(\langle r_0,\dots,r_n\rangle). \tag{5}$$

$$\mathrm{I}(c),\mathrm{I}(c') \to \mathrm{I}(\ulcorner c \oplus c'\urcorner) \qquad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{6}$$

$$\mathrm{I}(c),\mathrm{I}(x) \to \mathrm{I}(c \oplus x) \qquad \text{for each } c \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{7}$$

$$\mathrm{I}(c),\mathrm{I}(c' \oplus x) \to \mathrm{I}(\ulcorner c \oplus c'\urcorner \oplus x) \qquad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{8}$$

$$\mathrm{I}(c \oplus x),\mathrm{I}(c' \oplus x) \to \mathrm{I}(\ulcorner c \oplus c'\urcorner) \qquad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{9}$$

Figure 2: Rules of the theory T^+. We use the convention that $I(0 \oplus x)$ stands for $I(x)$.

Now, consider the Horn clauses induced by (6)–(9). For example, the set of Horn clauses (8) contains among others: $\mathrm{I}(a\oplus b),\mathrm{I}(b\oplus x) \to \mathrm{I}(a\oplus x)$ and $\mathrm{I}(b),\mathrm{I}(a\oplus x) \to \mathrm{I}((a\oplus b)\oplus x)$.

4.3 Proof of Theorem 1

In what follows, let T be a C-dominated Horn theory and b be a C-dominated message in normal form. Note that $\ulcorner b\urcorner = b$. The following lemma proves that our reduction is sound, i.e., that $T^+ \vdash b$ implies $T \vdash_{\oplus} b$.

Lemma 13. *If π is a syntactic derivation for b from T^+, then π is a derivation for b from T modulo XOR.*

Proof. Let π be a syntactic derivation for b from T^+. To prove the lemma it suffices to prove that each $\pi(i)$ can be obtained by a derivation modulo XOR from T and $\pi_{<i}$. If $\pi(i)$ is obtained from $\pi(j)$ and $\pi(k)$ for $j,k < i$, using one of the Horn clauses (6)–(9), then we can apply the $\oplus$-rule with $\pi(j)$ and $\pi(k)$ to obtain $\pi(j) \oplus \pi(i) \sim \pi(i)$.

Now, suppose that $\pi(i)$ is obtained using a Horn clause in (5) of the form $\ulcorner r_1\sigma\urcorner,\dots,\ulcorner r_n\sigma\urcorner \to \ulcorner r_0\sigma\urcorner$ for some Horn clause $(r_1,\dots,r_n \to r_0) \in T$ and some $\sigma \in \Sigma(\langle r_0,\dots,r_n\rangle)$. Hence, there exists a substitution θ and, for each $k \in \{1,\dots,n\}$, there exists $j < i$ such that $\pi(j) = \ulcorner r_k\sigma\urcorner\theta \sim (r_k\sigma)\theta = r_k(\sigma\theta)$. So, we can use the rule $r_1,\dots,r_n \to r_0$ to obtain $r_0(\sigma\theta) = (r_0\sigma)\theta \sim \ulcorner r_0\sigma\urcorner\theta = \pi(i)$. Note that $\ulcorner t\urcorner \sim t$ and if $t \sim t'$, then $t\sigma \sim t'\sigma$ for all terms t,t' and substitutions σ. □

To prove the completeness of our reduction, i.e., that $T \vdash_{\oplus} b$ implies $T^+ \vdash b$, we first prove the property of $\Sigma(t)$ mentioned before Definition 4. For this, we need the following definition.

Definition 5. Let t be a C-dominated term and θ be a C-dominated, ground substitution in normal form with $\mathrm{dom}(\theta) = \mathrm{var}(t)$. Let $\sigma = \sigma(t,\theta)$ be the substitution defined as follows. The domain of σ is the set of all variables that occur in some $s \in \mathcal{F}(t)$. Let x be such a variable. We define $\sigma(x)$ according to the following conditions, which have decreasing priority:

(a) If there exists $s \in \mathcal{F}(t)$ with $x \in \mathrm{var}(s)$ such that $s\theta \in \mathsf{C}^{\oplus}$, then $\sigma(x) = \theta(x)$.

(b) Otherwise, if $x \in \mathcal{F}(t)$ and $\theta(x) = c\oplus s'$, for $c \in \mathsf{C}^{\oplus}$ and some standard term s' not in $\mathsf{C}^{\oplus}$, then $\sigma(x) = c \oplus x$. (Note that $c \neq 0$ since $\theta(x)$ is in normal form.)

(c) Otherwise, $\sigma(x) = x$. (Note that in this case we know that $\theta(x)$ is some standard term not in $\mathsf{C}^{\oplus}$ if $x \in \mathcal{F}(t)$.)

Equipped with this definition, we show (see the appendix) the property of $\Sigma(t)$ mentioned before Definition 4.

Lemma 14. *Let t be a C-dominated term and θ be a C-dominated, ground substitution in normal form with $\mathrm{dom}(\theta) = \mathrm{var}(t)$. Then, $\sigma = \sigma(t,\theta) \in \Sigma(t)$ and there exists a substitution θ' such that $\theta = \sigma\theta'$, i.e., $\theta(x) = \sigma(x)\theta'$ for every $x \in \mathrm{dom}(\theta)$, and $\ulcorner t'\theta\urcorner = \ulcorner t'\sigma\urcorner\theta'$ for every subterm t' of t.*

We can now show the completeness of our reduction.

Lemma 15. *If π is a C-dominated derivation for b from T modulo XOR, then $\ulcorner\pi\urcorner$ is a syntactic derivation for b from T^+.*

Proof. We show that every $\ulcorner\pi(i)\urcorner$ can be derived syntactically from T^+ and $\ulcorner\pi_{<i}\urcorner$. Two cases are distinguished:

Case 1: $\pi(i)$ is obtained from $\pi(j) = \mathrm{I}(t)$ and $\pi(k) = \mathrm{I}(s)$, for $j,k < i$, using the $\oplus$-rule. In that case $\pi(i) \sim \mathrm{I}(t\oplus s)$. By assumption t, s, and $t \oplus s$ are C-dominated, and hence, $\ulcorner t\urcorner$, $\ulcorner s\urcorner$, $\ulcorner t\oplus s\urcorner$ are either normalized standard terms not in $\mathsf{C}^{\oplus}$, terms in $\mathsf{C}^{\oplus}_{\mathsf{norm}}$, or terms of the form $c\oplus u$ for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and a normalized standard term $u \notin \mathsf{C}^{\oplus}$, respectively. However, it is not the case that $\ulcorner t\urcorner = c\oplus u$ or $\ulcorner t\urcorner = u$ and $\ulcorner s\urcorner = u' \notin \mathsf{C}^{\oplus}$ or $\ulcorner s\urcorner = c' \oplus u'$ with $u \neq u'$ since otherwise $\ulcorner t \oplus s\urcorner$ would not be C-dominated. Now, it is easy to see that $\oplus$-rule can be simulated by one of the Horn clauses (6)–(9).

Case 2: $\pi(i)$ is obtained using some C-dominated rule $(r_1,\dots,r_n \to r_0) \in T$ and a ground substitution θ. Since π is C-dominated, by Lemma 6 and 3 we may assume that θ is C-dominated. Since π is a derivation modulo XOR, we may also assume that θ is in normal form. We have that $\pi(i) \sim r_0\theta$ and there exist $j_1,\dots,j_n < i$ such that $\pi(j_k) \sim r_k\theta$, for all $k \in \{1,\dots,n\}$.

Let $\sigma = \sigma(\langle r_0,\dots,r_n\rangle,\theta)$ and let θ' be as specified in Lemma 14. By Lemma 14, $\sigma \in \Sigma(\langle r_0,\dots,r_n\rangle)$. Now, to obtain $\ulcorner \pi(i) \urcorner$, we can use the rule $\rho = (\ulcorner r_1\sigma \urcorner, \ldots, \ulcorner r_n\sigma \urcorner \rightarrow \ulcorner r_0\sigma \urcorner) \in T^+$ with the substitution θ'. In fact, by Lemma 14, we have that $\ulcorner r_k\sigma \urcorner \theta' = \ulcorner r_k\theta \urcorner = \ulcorner \pi(j_k) \urcorner$ for all $k \in \{0, \ldots, n\}$, where $j_0 = 0$. (Recall that for C-dominated terms s and t with $s \sim t$, we have that $\ulcorner s \urcorner = \ulcorner t \urcorner$.) □

Now, from the above lemma and Proposition 1 it immediately follows that $T \vdash_\oplus b$ implies $T^+ \vdash b$.

5 Authentication

In the previous section, we showed how to reduce the derivation problem modulo XOR for C-dominated Horn theories to the syntactic derivation problem. While the derivation problem corresponds to the secrecy problem for cryptographic protocols w.r.t. an unbounded number of sessions, in this section, we will see that it is not hard to extend our result to authentication properties.

Authentication as Correspondence Assertions

Authentication properties are often expressed as *correspondence assertions* of the form $\mathsf{end}(x) \rightarrow \mathsf{begin}(x)$ where x describes the parameters on which the begin and end events should agree. This correspondence should be read as follows: If event $\mathsf{end}(x)$ has occurred, then also event $\mathsf{begin}(x)$. For example, $\mathsf{end}(a, b, n) \rightarrow \mathsf{begin}(a, b, n)$ could be interpreted as: If b thinks to have finished a run of a protocol with a in which the nonce n was used (in this case event $\mathsf{end}(a, b, n)$ occurred), then a has actually run a protocol with b in which n was used (in this case event $\mathsf{begin}(a, b, n)$ occurred). To check such correspondence assertions in the Horn theory based approach, roughly speaking, the protocol rules are augmented with atoms representing events of the form $\mathsf{begin}(x)$ and $\mathsf{end}(x)$ (see, e.g., [3] for details).

For our running example, this is illustrated in Figure 3. In (13), the end event indicates that b believes to have talked to a and the nonce $m(b, a, sid, x)$ was used in the interaction, where x is the nonce b believes to have received from a and sid is a session identifier. The parameters x and sid are added to the term representing the nonce in order to make the analysis more precise. In particular, the session identifier is added in order to make the correspondence stronger: The events should not only correspond on the names and the nonces used in the protocol run, but also on the session identifiers. Note that without the session identifier, correspondence of sessions would otherwise not be guaranteed since in the Horn theory based approach new protocol runs do not necessarily use completely fresh nonces. The begin event in (12) indicates that a just received the response from b and now outputs her response to b, where the begin event contains the nonce received from b.

We note that, strictly speaking, the Horn theory depicted in Figure 3 falls out of the class of Horn theories that we allow, not because of $\oplus$-linearity but because of the fact that the variable sid occurs on the right-hand side of a Horn clause but not on the left-hand side (see (10) and (11)). However, as we noted in Section 2, this assumption can easily be relaxed for variables that are supposed to be substituted only by C-dominated terms, which is the case for session identifiers.

Now, let T be a Horn theory model of a protocol and an intruder, i.e., T consists of a set of protocol rules (such as those in Figure 3), a set of initial intruder facts, and a set of intruder rules. Following Blanchet [3], we say that a (non-injective) correspondence assertion of the form $\mathsf{end}(x) \rightarrow \mathsf{begin}(x)$ is satisfied by T if

> for every finite set of messages B and every message $m_0 \notin \widehat{B}$, it holds that $T \cup \{\mathsf{begin}(m) \mid m \in B\} \nvdash_\oplus \mathsf{end}(m_0)$, (14)

where $\widehat{B} = \{t \mid \text{there exists } t' \in B \text{ and } t \sim t'\}$. In [3], this formulation (more precisely, a syntactic version, i.e., the XOR-free version) is somewhat implicit in a theorem which reduces correspondence assertions in process calculus to Horn theories. Blanchet then proposes a method for proving the syntactic version of (14) using ProVerif.

Extending Our Reduction to Correspondence Assertions

The following theorem extends our reduction presented in Section 4 to the problem of solving (14) *with* XOR. In fact, we show that if in (14) the (C-dominated) Horn theory T is replaced by T^+ (i.e., we can use the same reduction function as in Section 4), then derivation modulo XOR ($\vdash_\oplus$) can be replaced by syntactic derivation ($\vdash$). Now, the latter problem (the syntactic version of (14)) can be solved using ProVerif. Formally, we can prove:

Theorem 2. *Let T be a C-dominated Horn theory. Then, (14) holds iff for every finite set of messages B and every message $m_0 \notin B$, it holds that $T^+ \cup \{\mathsf{begin}(m) \mid m \in B\} \nvdash \mathsf{end}(m_0)$.*

The proof of this theorem requires some slight extension of Proposition 1, stated below, in which an injective version of Δ is used, i.e., $t \not\sim t'$ should imply that $\Delta(t) \not\sim \Delta(t')$. This is needed to guarantee that if $m_0 \notin \widehat{B}$, then $\Delta(m_0) \notin \widehat{\Delta(B)}$.

This can be achieved by fixing an *injective* function γ which takes a term to some term built from 0 and $\langle \cdot, \cdot \rangle$ (or any other function which the intruder can apply). We also add the fresh constant c_0 to the intruders knowledge. Now, for a bad term $t = c \oplus t_1 \oplus \cdots \oplus t_n$, we define $\Delta(t) = c \oplus \tilde{t_1} \oplus \cdots \oplus \tilde{t_n} \oplus \{\gamma(t)\}_{\mathsf{c}_0}$. The important property of

$$\mathrm{I}(\{n(a,b,\mathit{sid}),a\}_{\mathsf{pub}(k_b)}) \qquad \text{for every } a \in \mathsf{H},\ b \in \mathsf{P} \tag{10}$$

$$\mathrm{I}(\{x,a\}_{\mathsf{pub}(k_b)}) \;\to \mathrm{I}(\{m(b,a,\mathit{sid},x), x \oplus b\}_{\mathsf{pub}(k_a)}) \qquad \text{for every } b \in \mathsf{H},\ a \in \mathsf{P} \tag{11}$$

$$\mathsf{begin}(a,b,y),\ \mathrm{I}(\{y, n(a,b,\mathit{sid}) \oplus b\}_{\mathsf{pub}(k_a)}) \;\to \mathrm{I}(\{y\}_{\mathsf{pub}(k_b)}) \qquad \text{for every } a \in \mathsf{H},\ b \in \mathsf{P} \tag{12}$$

$$\mathrm{I}(\{(x,a)\}_{\mathsf{pub}(k_b)}),\ \mathrm{I}(\{m(b,a,\mathit{sid},x)\}_{\mathsf{pub}(k_b)}) \;\to \mathsf{end}(a,b,m(b,a,\mathit{sid},x)) \qquad \text{for every } b \in \mathsf{H},\ a \in \mathsf{P} \tag{13}$$

Figure 3: Rules for authentication (*sid* is a variable intended to range over session identifiers).

$\{\gamma(t)\}_{\mathsf{c}_0}$ is that the intruder can derive this message and that it is unique for every term t.

Proposition 2. *Let T be a C-dominated Horn theory, B be a finite set of facts, and a be a fact. If $T \cup B \vdash_\oplus a$, then there exists a C-dominated derivation for $\Delta(a)$ from $T \cup \Delta(B)$ modulo XOR.*

The proof of this proposition is very similar to the one of Proposition 1. Only minor modifications are necessary.

Now, to prove Theorem 2, it suffices to show that the following conditions are equivalent, for a C-dominated theory T:

(i) there exist a finite set of messages B and a message $m_0 \notin \widehat{B}$ such that $T \cup \{\mathsf{begin}(m) \mid m \in B\} \vdash_\oplus \mathsf{end}(m_0)$

(ii) there exist a finite set of C-dominated messages B and a C-dominated message $m_0 \notin \widehat{B}$ such that $T \cup \{\mathsf{begin}(m) \mid m \in B\} \vdash_\oplus \mathsf{end}(m_0)$.

(iii) there exist a finite set of C-dominated messages B and a C-dominated message $m_0 \notin B$ such that $T^+ \cup \{\mathsf{begin}(m) \mid m \in B\} \vdash \mathsf{end}(m_0)$.

(iv) there exist a finite set of messages B and a message $m_0 \notin B$ such that $T^+ \cup \{\mathsf{begin}(m) \mid m \in B\} \vdash \mathsf{end}(m_0)$.

Proof. The implication (i)⇒(ii) follows from Proposition 2 and by the fact that Δ is injective; (ii)⇒(iii) is given by Theorem 1 (we use the fact that $T \cup \{\mathsf{begin}(m) \mid m \in B\}$ is C-dominated and the fact that $(T \cup \{\mathsf{begin}(m) \mid m \in B\})^+ = T^+ \cup \{\mathsf{begin}(m) \mid m \in \ulcorner B \urcorner\}$); (iii)⇒(iv) is trivial; finally, (iv)⇒(i) is given by Lemma 13. □

6 Implementation and Experiments

We have implemented our reduction, and together with ProVerif, tested it on a set of protocols which employ the XOR operator (see [17] for the implementation). In this section, we report on our implementation and the experimental results.

6.1 Implementation

We have implemented our reduction function in SWI prolog (version 5.6.14). Our implementation essentially takes a Horn theory as input. More precisely, the input consists of (1) a declaration of all the functor symbols used in the protocol and by the intruder, (2) the initial intruder facts as well as the protocol and intruder rules, except for the $\oplus$-rule, which is assumed implicitly, (3) a statement which defines a secrecy or authentication goal. Moreover, options that are handed over to ProVerif may be added.

Our implementation then first checks whether the given Horn theory, say T, (part (2) of the input) is $\oplus$-linear. If it is not, an error message is returned. If it is, a set C is computed such that the Horn theory is C-dominated. Recall that such a set always exists if the Horn theory is $\oplus$-linear. It is important to keep C as small as possible, in order for the reduction to be more efficient. Once C is computed, the reduction function as described in Section 4 is applied to T, i.e., T^+ is computed. Now, T^+ together with the rest of the original input is passed on to ProVerif. This tool then does the rest of the work, i.e., it checks the goals for T^+. This is possible since, due the reduction, the XOR operator in T^+ can now be considered to be an operator without any algebraic properties.

Our implementation does not follow the construction of the reduction function described in Section 4 precisely, in order to produce an output that is optimized for ProVerif (but still equivalent): a) While terms of the form $c \oplus t$, with $c \in \mathsf{C}^\oplus$, $t \notin \mathsf{C}^\oplus$ are represented by $\mathtt{xor(c,t)}$, terms $a \oplus b \in \mathsf{C}^\oplus_{\mathsf{norm}}$ are represented by $\mathtt{xx(a,b)}$. This representation prevents some unnecessary unifications between terms. However, it is easy to see that with this representation, the proofs of soundness and completeness of our reduction still go through. The basic reason is that terms in $\mathsf{C}^\oplus_{\mathsf{norm}}$ can be seen as constants. b) For the Horn clauses in Figure 2, (6)–(9), we do not produce copies for every choice of $c, c' \in \mathsf{C}^\oplus_{\mathsf{norm}}$. Instead, we use a more compact representation by introducing auxiliary predicate symbols. For example, the family of Horn clauses in (8) is represented as follows: $\mathtt{xtab(x,y,z)}, \mathtt{I(y)}, \mathtt{I(xor(x,t))} \to \mathtt{I(xor(z,t))}$, where the facts $\mathtt{xtab}(\mathtt{c},\mathtt{c}',\ulcorner \mathtt{c} \oplus \mathtt{c}' \urcorner)$ for every $c, c' \in \mathsf{C}^\oplus_{\mathsf{norm}}$

protocol	correct	reduction time	ProVerif time
NSL$_\oplus$	no	0.02s	0.006s
NSL$_\oplus$-fix	yes	0.04s	0.09s
NSL$_\oplus$-auth-A	no	0.03s	0.16s
NSL$_\oplus$-auth-A-fix	yes	0.03s	0.02s
NSL$_\oplus$-auth-B	yes	0.04s	0.5s
SK3	yes	0.05s	0.3s
RA	no	0.05s	0.17s
RA-fix	yes	0.05s	0.27s
CCA-0	no	0.15s	109s
CCA-1A	yes	0.06s	0.7s
CCA-1B	yes	0.07s	1.3s
CCA-2B	yes	0.14s	7.1s
CCA-2C	yes	0.15s	58.0s
CCA-2E	yes	0.07s	1.42s

Figure 4: Experimental Results.

are added to the Horn theory given to ProVerif.

6.2 Experiments

We applied our method to a set of ($\oplus$-linear) protocols. The results, obtained by running our implementation on a 2,4 Ghz Intel CoreTM 2 Duo E6700 processor with 2GB RAM, are depicted in Figure 4, where we list both the time of the reduction and the time ProVerif needed for the analysis of the output of the reduction. We note that except for certain versions of the CCA protocol, the other protocols listed in Figure 4 are out of the scope of the implementation in [14], the only other implementation that we know of for cryptographic protocol analysis w.r.t. an unbounded number of sessions that takes XOR into account. As mentioned in the introduction, the method in [14] is especially tailored to the CCA protocol. It can only deal with symmetric encryption and the XOR operator, but, for example, cannot deal with protocols that use public-key encryption or pairing. Let us discuss the protocols and settings that we analyzed in more detail.

By NSL$_\oplus$ we denote our running example. Since there is an attack on this protocol, we also propose a fix NSL$_\oplus$-fix in which the message $\{\!|\langle M, N \oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)}$ is replaced by $\{\!|\langle M, h(\langle N, M\rangle) \oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)}$ for a hash function $h(\cdot)$. We analyze both authentication and secrecy properties for these ($\oplus$-linear) protocols.

The ($\oplus$-linear) protocol SK3 [18] is a key distribution protocol for smart cards, which uses the XOR operator. RA denotes an ($\oplus$-linear) group protocol for key distribution [6]. Since there is a known attack on this protocol, we proposed a fix: a message $k_{A,B} \oplus h(\langle \mathsf{key}(A), N\rangle)$ sent by the key distribution server to A is replaced by $k_{A,B} \oplus h(\langle \mathsf{key}(A), \langle N, B\rangle\rangle)$.

CCA stands for Common Cryptographic Architecture (CCA) API [1] as implemented on the hardware security module IBM 4758 (an IBM cryptographic coprocessor). The CCA API is used in ATMs and mainframe computers of many banks to carry out PIN verification requests. It accepts a set of commands, which can be seen as receive-send-actions, and hence, as cryptographic protocols. The only key stored in the security module is the master key $\textsc{km}$. All other keys are kept outside of the module in the form $\{k\}_{\textsc{km}\oplus type}$, where $type \in \{\textsc{data}, \textsc{imp}, \textsc{exp}, \textsc{pin}\}$ denotes the type of the key, where each type is some fixed constant. The commands of the CCA API include the following: Commands for encrypting/decrypting data using data keys. Commands to export/import a key to/from another security module. This is done by encrypting/decrypting the key by a key-encryption-key.

In Figure 5, we model the most important commands of the CCA API (see also [14]) in terms of Horn clauses. (*Encipher*) and (*Decipher*) are used to encrypt/decrypt data by data keys. (*KeyExport*) is used to export a key to another security module by encrypting it under a key-encryption-key, with (*KeyImport*) being the corresponding import command. The problem is to make the same key-encryption-key available in different security modules. This is done by a secret sharing scheme using the commands (*KeyPartImp-First*)–(*KeyPartImp-Last*), where $\textsc{kp}$ is a type (a constant) which stands for "key part", kek is obtained as $k1 \oplus k2 \oplus k3$, and each ki, $i \in \{1, 2, 3, \}$, is supposed to be known by only one individual. (*KeyTranslate*) is used to encrypt a key under a different key-encryption-key.

We note that some of the Horn clauses in Figure 5, namely (*KeyPartImp-Middle*) and (*KeyPartImp-Last*), are not linear. Fortunately, one can apply a standard unfolding technique for Horn clauses together with straightforward simplifications to obtain an *equivalent* Horn theory with only $\oplus$-linear rules.

There are several known attacks on the CCA API, which concern the key-part-import process. One attack is by Bond [5]. As a result of this attack the intruder is able to obtain PINs for each account number by performing data encryption on the security module. A stronger attack was found by IBM and is presented in [8] where the intruder can obtain a PIN derivation key, and hence, can obtain PINs even without interacting with the security module. However, the IBM attack depends on key conjuring [14], and hence, is harder to carry out. Using our implementation (together with ProVerif) and the configuration denoted by CCA-0 in Figure 4, we found a new attack which achieves the same as the IBM attack, but is more efficient as it does not depend on key conjuring. Our attack is presented at the end of this section.

In response to the attacks reported in [5], IBM proposed two recommendations.

Recommendation 1. As mentioned, the attacks exploit

$$\begin{aligned}
I(x),\ I(\{k\}_{\mathrm{KM}\oplus\mathrm{DATA}}) &\rightarrow I(\{x\}_k) && (\mathit{Encipher})\\
I(\{x\}_k),\ I(\{k\}_{\mathrm{KM}\oplus\mathrm{DATA}}) &\rightarrow I(x) && (\mathit{Decipher})\\
I(\{k\}_{\mathrm{KM}\oplus \mathit{type}}),\ I(\mathit{type}),\ I(\{\mathit{kek}\}_{\mathrm{KM}\oplus\mathrm{EXP}}) &\rightarrow I(\{k\}_{\mathit{kek}\oplus\mathit{type}}) && (\mathit{KeyExport})\\
I(\{k\}_{\mathit{kek}\oplus\mathit{type}}),\ I(\mathit{type}),\ I(\{\mathit{kek}\}_{\mathrm{KM}\oplus\mathrm{IMP}}) &\rightarrow I(\{k\}_{\mathrm{KM}\oplus\mathit{type}}) && (\mathit{KeyImport})\\
I(k1), I(\mathit{type}) &\rightarrow I(\{k1\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus\mathit{type}}) && (\mathit{KeyPartImp\text{-}First})\\
I(k2), I(\{x\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus\mathit{type}}), I(\mathit{type}) &\rightarrow I(\{x\oplus k2\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus\mathit{type}}) && (\mathit{KeyPartImp\text{-}Middle})\\
I(k3), I(\{y\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus\mathit{type}}), I(\mathit{type}) &\rightarrow I(\{y\oplus k3\}_{\mathrm{KM}\oplus\mathit{type}}) && (\mathit{KeyPartImp\text{-}Last})\\
I(\{k\}_{\mathit{kek}_1\oplus\mathit{type}}),\ I(\mathit{type}),\ I(\{\mathit{kek}_1\}_{\mathrm{KM}\oplus\mathrm{IMP}}),\ I(\{\mathit{kek}_2\}_{\mathrm{KM}\oplus\mathrm{EXP}}) &\rightarrow I(\{k\}_{\mathit{kek}_2\oplus\mathit{type}}) && (\mathit{KeyTranslate})
\end{aligned}$$

Figure 5: CCA API , where KM denotes a constant (the key master stored in the cryptographic coprocessor), *type* is a constant that ranges over the constants in $\{\mathrm{DATA}, \mathrm{IMP}, \mathrm{EXP}, \mathrm{PIN}\}$, and all other symbols (x, y, k, ...) are variables.

problems in the key-part-import process. To prevent these problems, one IBM recommendation is to replace this part by a public-key setting. However, as shown in [14], further access control mechanisms are needed, which essentially restrict the kind of commands certain roles may perform. Two cases, which correspond to two different roles, are considered, and are denoted CCA-1A and CCA-1B in Figure 4. We note that the Horn theories that correspond to these cases are $\oplus$-linear, and hence, our tool can be applied directly, no changes are necessary; not even the transformations mentioned above. Since public-key encryption (and pairing) cannot be directly handled by the tool presented by Cortier et al. [14], Cortier et al. had to modify the protocol in an ad hoc way, which is not guaranteed to yield an equivalent protocol. This is also why the runtimes of the tools cannot be compared directly.

Recommendation 2. Here additional access control mechanisms are assumed which ensure that no single role is able to mount an attack. We analyzed exactly the same subsets of commands as the ones in [14]. These cases are denoted CCA-2B, -2C, and -2E in Figure 4, following the notation in [14]. The runtimes obtained in [14] are comparable to ours: 333s for CCA-2B, 58s for -2C, and 0.03s for -2E.

Our Attack. As we noted before, our tool found an attack which—according to our knowledge—has not been discovered before. This attack uses the same assumptions as Bond's attack in terms of the role played by the intruder and his knowledge. As in the IBM attack, we use the fact that 0 is the default value for DATA.

Our attack does not use key conjuring, and hence, is easier to carry out than the IBM attack. As a result of the attack, the intruder obtains a pin derivation key in clear (like in the IBM attack).

In the attack we assume that a new key-encryption-key *kek* needs to be imported, using the three-part key import commands (*KeyPartImp-First*)–(*KeyPartImp-Last*), which means that $\mathit{kek} = k1 \oplus k2 \oplus k3$, where $k1$, $k2$, $k3$ are the shares known by three different individuals.

The key *kek* is then used to import a new pin-derivation key *pdk* to the security module, in the form

$$\{\mathit{pdk}\}_{\mathit{kek}\oplus\mathrm{PIN}}. \tag{15}$$

We assume that this message can be seen by the attacker and that the attacker is the third participant of the process of importing *kek*. In particular, the attacker can perform (*KeyPartImp-Last*), knows the value $k3$, and obtains the message

$$\{k1 \oplus k2\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus\mathrm{IMP}}. \tag{16}$$

Now we describe the steps of the attack. After the intruder receives (16), he uses (*KeyPartImp-Last*) with $k3 \oplus \mathrm{PIN}$ instead of $k3$. In this way he obtains

$$\{\mathit{kek} \oplus \mathrm{PIN}\}_{\mathrm{KM}\oplus\mathrm{IMP}} \tag{A1}$$

He uses the same command again, this time with $k3 \oplus \mathrm{PIN} \oplus \mathrm{EXP}$, obtaining:

$$\{\mathit{kek} \oplus \mathrm{PIN} \oplus \mathrm{EXP}\}_{\mathrm{KM}\oplus\mathrm{IMP}} \tag{A2}$$

Next, when *pdk* is imported, the intruder uses (*KeyImport*) twice: The first time with input (A1), (15), and $\mathit{type} = \mathrm{DATA} = 0$, resulting in the message

$$\{\mathit{pdk}\}_{\mathrm{KM}\oplus\mathrm{DATA}}. \tag{A3}$$

The second time with input (A2), (15), and $\mathit{type} = \mathrm{EXP}$, resulting in the message

$$\{\mathit{pdk}\}_{\mathrm{KM}\oplus\mathrm{EXP}}. \tag{A4}$$

Now, using (*KeyExport*) with input (A3), (A4), and $\mathit{type} = \textsc{data} = 0$, the attacker obtains

$$\{\mathit{pdk}\}_{\mathit{pdk}\oplus\textsc{data}} = \{\mathit{pdk}\}_{\mathit{pdk}}. \qquad \text{(A5)}$$

Finally, using (*Decipher*) with input (A5) and (A3), the attacker obtains the clear value of *pdk*, which can be used to obtain the PIN for any account number: Given an account number, the corresponding PIN is derived by encrypting the account number under *pdk*.

A Proofs for Section 3

In what follows we will use the following notation: $t \mathrel{\hat{=}}_{AC} t'$ if t and t' are coincide up to transformation modulo AC, with standard terms kept unchanged. For example, $(a \oplus \langle a \oplus b, b\rangle) \oplus b \mathrel{\hat{=}}_{AC} (a \oplus b) \oplus \langle a \oplus b, b\rangle \not\mathrel{\hat{=}}_{AC} (a \oplus b) \oplus \langle b \oplus a, b\rangle$.

Proof of Lemma 4.

Assume that r' is a complete bad subterm of $r\theta$. We proceed by structural induction on r and consider the following cases:

- $r = x$ is a variable: Because θ is $\oplus$-reduced, so is $\theta(x)$. So, since r' is a subterm of $\theta(x)$ and $\theta(x) \sim t$, Lemma 3 implies that there exists a complete bad subterm t' of t with $t' \sim r'$.

- $r = f(r_1, \ldots, r_n)$, for $f \neq \oplus$: In this case, t is of the form $f(t_1, \ldots, t_n)$ with $t_i \sim r_i\theta$. Since $r\theta$ is not bad, r' is a subterm of $r_i\theta$ for some $i \in \{1, \ldots, n\}$. By the induction hypothesis, there exists a complete bad subterm t' of t_i (and thus, of t) with $t' \sim r'$.

- $r = c$, for $c \in \mathsf{C}^{\oplus}$: We have that $r\theta = r$. Since r is C-dominated it follows that c does not contain complete bad subterms. Hence, nothing is to show.

- $r \mathrel{\hat{=}}_{AC} c \oplus r''$ with $c \in \mathsf{C}^{\oplus}$ and $r'' \notin \mathsf{C}^{\oplus}$ standard, but not a variable: The case that $r' = r\theta$ cannot occur since this term is not a bad term. Since r is C-dominated, c does not contain a complete bad subterm. Hence, r' cannot be a subterm of $c\theta = c$. So r' is a subterm of $r''\theta$.

  Let $s \sim r''\theta$, for some $\oplus$-reduced term $s \in \mathsf{C}^{\oplus}$. So, we have that $t \sim c \oplus s$. Since r'', as a proper subterm of r, is C-dominated, from the fact that r' is a complete bad subterm of $r''\theta$ it follows by the induction hypothesis that there exists a complete bad subterm t' of s with $r' \sim t'$. Now, since c is C-dominated (because by assumption r is), and hence, c does not contain complete bad subterms, it follows that t' occurs as a subterm in t.

- $r \mathrel{\hat{=}}_{AC} c \oplus x$, for $c \in \mathsf{C}^{\oplus}$ and a variable x: Assume that $\theta(x) \sim c' \oplus t_1 \oplus \cdots \oplus t_n$ with $n \geq 0$, $c' \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$. First assume that $r' = r\theta$, which implies that $n > 1$. Then we can set $t' = t$ since $t' = t \sim r\theta = r'$. Otherwise, since r is C-dominated, it follows that c does not contain a complete bad subterm. Hence, r' is a complete bad subterm of c' or there exists i such that r' is a complete bad subterm of t_i. In any case, this term, let us call it t'', does not coincide with any standard term c_i with $c = c_1 \oplus \ldots \oplus c_k$ because these terms do not contain complete bad subterms. Hence, t'' is equivalent to some term t' in t. Thus, there exists a complete bad subterm t' of t with $r' \sim t'$. □

Proof of Lemma 5.

We proceed by structural induction on s:

- *$s = x$ is a variable*: We can set $t' = t$.

- *s is standard*: Then $s \neq t$, and thus, for one of the direct subterms s' of s, $s'\theta$ has to contain t as a complete subterm. By the induction hypothesis, there exists a variable $x \in \mathrm{var}(s') \subseteq \mathrm{var}(s)$ such that $\theta(x)$ contains a complete bad subterm t' with $t' \simeq_{\mathsf{C}} t$.

- *$s \in \mathsf{C}^{\oplus}$*: This case is not possible, since $s = s\theta$ is C-dominated, and hence, cannot contain a complete bad subterm.

- *$s \mathrel{\hat{=}}_{AC} c \oplus s'$, where $c \in \mathsf{C}^{\oplus}$ and $s' \notin \mathsf{C}^{\oplus}$ is standard, but not a variable*: Then, $t \neq s\theta$ since $s\theta$ is not a bad term. Moreover, c is C-dominated (since it belongs to s), and hence, cannot have t as a subterm. Hence, t must be a subterm of $s'\theta$ and we can use the induction hypothesis.

- *$s \mathrel{\hat{=}}_{AC} c \oplus x$, for $c \in \mathsf{C}^{\oplus}$ and a variable x*: If $t \sim (c \oplus x)\theta$, we can choose $t' = \theta(x)$, since $t' \simeq_{\mathsf{C}} t$. Otherwise, since c is $\mathsf{C}^{\oplus}$-dominated, and hence, does not contain complete bad subterms, it follows that t is a subterm of $\theta(x)$. Hence, we can choose $t' = t$. □

B Proofs for Section 4

Proof of Lemma 12.

We start with showing that matching of C-dominated terms modulo XOR yields a uniquely determined matcher modulo XOR, if any, and this matcher can be computed in polynomial time.

Claim 1. Let s be a C-dominated term and t be a ground term. Then, the matcher of s against t is uniquely determined modulo XOR, i.e., if $s\theta \sim t$ and $s\theta' \sim t$ for substitutions θ and θ', then $\theta(x) \sim \theta'(x)$ for every $x \in \mathrm{var}(s)$.

Moreover, the matcher of s against t can be computed in polynomial time in the size of s and t.

Proof. We show how to compute the unique (modulo XOR) matcher of s against t. The computed matcher will be in normal form. First, for substitutions σ_1 and σ_2 we define $\sigma_1 \sqcup \sigma_2$ as $\sigma_1 \cup \sigma_2$ if for each $x \in \mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2)$ we have that $\sigma_1(x) = \sigma_2(x)$. Otherwise, $\sigma_1 \sqcup \sigma_2$ is undefined.

We obtain the matcher σ of s against t recursively as follows. We can assume that both s and t are in normal form (one can transform a term t into its normal form $\ulcorner t \urcorner$ in polynomial time)[2]. We consider the following cases:

1. $s = x$ is a variable: Then $\sigma = \{t/x\}$.
2. s is a ground term: Then $\sigma = \emptyset$ if $s = t$. Otherwise, the matcher does not exist.
3. $s = c \oplus s'$, for ground c and nonground, standard s': Then σ is the matcher of s' against the term $\ulcorner c \oplus t \urcorner$.
4. $s = f(s_1, \ldots, s_n)$, for $f \neq \oplus$, non ground: If $t = f(t_1, \ldots, t_n)$, we take $\sigma = \sigma_1 \sqcup \cdots \sqcup \sigma_n$, where σ_i, for $i \in \{1, \ldots, n\}$, is the matcher of s_i against t_i. Otherwise, i.e. if such a σ does not exist, the matcher does not exist.

It is easy to show that this algorithm computes a matcher of s against t, if it exists, and moreover, that this matcher is unique. □

Now, we are ready to prove Lemma 12: The domain of every substitution in $\Sigma(t)$ is polynomial, since it is a subset of $\mathrm{var}(t)$. Hence, it suffices to show that for every variable in the domain there are only exponentially many possible values and these values can be computed effectively. This is clear for the case (i) and (ii) in Definition 4, as $\mathsf{C}^{\oplus}_{\mathsf{norm}}$ is bounded exponentially (in the size of C).

As for case (iii), let s, x and θ be given as in this case. Note that s is C-dominated. Hence, θ is the unique matcher of s against some $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$. Because θ can be computed from s and c in polynomial time and, moreover, both s and c range over exponentially bounded sets (in fact, $\mathcal{F}(t)$ is polynomial and $\mathsf{C}^{\oplus}_{\mathsf{norm}}$ is exponential), the claim of the lemma follows.

Proof of Lemma 14.

Let t and θ be given as in the lemma. By construction, it is easy to see that $\sigma = \sigma(t, \theta) \in \Sigma(t)$. It is also easy to see that there exists θ' such that $\theta = \sigma\theta'$ and the domain of θ' is the set of all variables that occur in some $\sigma(x)$ for $x \in \mathrm{dom}(x)$. Note that θ' is uniquely determined. Let t' be a subterm of t. We need to show that $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner \theta'$. We proceed by structural induction on t'.

First, suppose that $t' \in \mathrm{var}(t)$: Let $x = t'$. We distinguish the following cases:

(a) If $\sigma(x)$ was defined according to Definition 5, (a), then $\sigma(x) = \theta(x)$. It follows that $\ulcorner x\theta \urcorner = \ulcorner x\sigma \urcorner \theta'$.

(b) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (b), then $x \in \mathcal{F}(t)$, $\theta(x) = c \oplus s'$, for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and some normalized standard term s' not in $\mathsf{C}^{\oplus}$, and $\sigma(x) = c \oplus x$. It follows that $\theta'(x) = s'$ and $\ulcorner x\sigma \urcorner \theta' = \ulcorner c \oplus x \urcorner \theta' = (c \oplus x)\theta' = c \oplus s' = \ulcorner c \oplus s' \urcorner = \ulcorner x\theta \urcorner$.

(c) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (c), then $\sigma(x) = x$ and $\theta'(x) = \theta(x)$. Since $\theta(x)$ is normalized, it follows that $\ulcorner x\theta \urcorner = \ulcorner x\sigma \urcorner \theta'$.

Second, suppose that $t' = f(t_1, \ldots, t_n)$, for $f \neq \oplus$: By the induction hypothesis, it follows that $\ulcorner t'\theta \urcorner = f(\ulcorner t_1\theta \urcorner, \ldots, \ulcorner t_n\theta \urcorner) = f(\ulcorner t_1\sigma \urcorner \theta', \ldots, \ulcorner t_n\sigma \urcorner \theta') = \ulcorner t'\sigma \urcorner \theta'$.

If we suppose that $t' \sim c$, for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$, then it immediately follows that $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner \theta'$.

Now, suppose that $t' \sim c \oplus x$, for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$: We distinguish the following cases:

(a) If $\sigma(x)$ was defined according to Definition 5, (a), then $\sigma(x) = \theta(x)$. It follows that $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner \theta'$.

(b) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (b), then $x \in \mathcal{F}(t)$, $\theta(x) = c' \oplus s'$, for $c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and some normalized standard term s' not in $\mathsf{C}^{\oplus}$, and $\sigma(x) = c' \oplus x$. It follows that $\theta'(x) = s'$ and $\ulcorner t'\sigma \urcorner \theta' = \ulcorner c \oplus c' \oplus x \urcorner \theta' = \ulcorner c \oplus c' \urcorner \oplus x\theta' = \ulcorner c \oplus c' \urcorner \oplus s' = \ulcorner c \oplus c' \oplus s' \urcorner = \ulcorner t'\theta \urcorner$.

(c) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (c), then $\sigma(x) = x$ and $\theta'(x) = \theta(x)$. Since $x \in \mathcal{F}(t)$ and items (a) and (b) of Definition 5 do not hold, $\theta'(x)$ is a normalized standard term not in $\mathsf{C}^{\oplus}_{\mathsf{norm}}$. It follows that $\ulcorner t'\theta \urcorner = \ulcorner c \oplus \theta(x) \urcorner = c \oplus \theta(x) = \ulcorner t'\sigma \urcorner \theta'$.

Finally, suppose that $t' \sim c \oplus s$, for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and a C-dominated, standard subterm s of t' with $s \notin \mathsf{C}^{\oplus}$ and $s \notin \mathrm{var}(t)$: We distinguish the following cases:

(a) If $s\theta \in \mathsf{C}^{\oplus}$, then $\sigma(x)$, for $x \in \mathrm{var}(s)$, was defined according to Definition 5, (a) since $s \in \mathcal{F}(t)$. Hence, $\sigma(x) = \theta(x)$ for all $x \in \mathrm{var}(s)$, and thus $s\sigma$ is ground and $s\sigma = s\theta$. It follows that $\ulcorner t'\theta \urcorner = \ulcorner c \oplus s\theta \urcorner = \ulcorner x \oplus s\sigma \urcorner = \ulcorner x \oplus s\sigma \urcorner \theta' = \ulcorner t'\sigma \urcorner \theta'$.

(b) Otherwise, if $s\theta \notin \mathsf{C}^{\oplus}$, by the induction hypothesis it follows that $\ulcorner s\theta \urcorner = \ulcorner s\sigma \urcorner \theta'$. We have also that $s\sigma$ is not in $\mathsf{C}^{\oplus}$ (otherwise, $s\theta$ would be also in $\mathsf{C}^{\oplus}$). Moreover, since $s\theta \notin \mathsf{C}^{\oplus}$, we obtain that $\ulcorner t'\theta \urcorner = c \oplus \ulcorner s\theta \urcorner = c \oplus \ulcorner s\sigma \urcorner \theta' = \ulcorner (c \oplus s)\sigma \urcorner \theta' = \ulcorner t'\sigma \urcorner \theta'$. □

References

[1] CCA Basic Services Reference and Guide: CCA Basic Services Reference and Guide, 2003. Available at `http://www-03.ibm.com/security/cryptocards/pdfs/bs327.pdf`.

[2] So far, we defined $\ulcorner \cdot \urcorner$ only for C-dominated terms. Now, we need to extend the definition of $\ulcorner \cdot \urcorner$ to work for all terms. Such a extension is straightforward. So we skip it.

[2] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW-14)*, pages 82–96. IEEE Computer Society, 2001.

[3] B. Blanchet. Automatic verification of correspondences for security protocols, 2008. Report arXiv:0802.3444v1. Available at `http://arxiv.org/abs/0802.3444v1`.

[4] B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.

[5] M. Bond. Attacks on cryptoprocessor transaction sets. In *CHES*, volume 2162 of *LNCS*, pages 220–234. Springer, 2001.

[6] J. Bull and D. Otway. The authentication protocol. Technical Report `DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03`, Defence Research Agency, Malvern, UK, 1997.

[7] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proceedings of the Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 261–270. IEEE, Computer Society Press, 2003.

[8] J. Clulow. The design and analysis of cryptographic APIs for security devices, 2003. Master's thesis, University of Natal, Durban.

[9] H. Comon-Lundh and V. Cortier. New Decidability Results for Fragments of First-order Logic and Application to Cryptographic Protocols. In *Proceedings of the 14th Internatioinal Conference on Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2003.

[10] H. Comon-Lundh and V. Cortier. Security properties: two agents are sufficient. *Sci. Comput. Program.*, 50(1-3):51–71, 2004.

[11] H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In *RTA*, volume 3467 of *LNCS*, pages 294–307. Springer, 2005.

[12] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proceedings of the Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 271–280. IEEE, Computer Society Press, 2003.

[13] V. Cortier, S. Delaune, and G. Steel. A formal theory of key conjuring. In *20th IEEE Computer Security Foundations Symposium (CSF'07)*, pages pages 79–93. IEEE Comp. Soc. Press, 2007.

[14] V. Cortier, G. Keighren, and G. Steel. Automatic Analysis of the Security of XOR-Based Key Management Schemes. In *Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007)*, volume 4424 of *LNCS*, pages 538–552. Springer, 2007.

[15] R. Küsters and T. Truderung. On the Automatic Analysis of Recursive Security Protocols with XOR. In W. Thomas and P. Weil, editors, *Proceedings of the 24th Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, volume 4393 of *LNCS*, pages 646–657. Springer, 2007.

[16] R. Küsters and T. Truderung. Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*. ACM Press, 2008.

[17] R. Küsters and T. Truderung. Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach. Implementation, 2008. Available at `http://infsec.uni-trier.de/software/KuestersTruderung-XORPROVERIF-2008.zip`.

[18] V. Shoup and A. Rubin. Session key distribution using smart cards. In *Andances in Cryptology, EUROCRYPT*, volume 1070/1996 of *LNCS*, pages 321–331. Springer, 1996.

[19] G. Steel. Deduction with xor constraints in security api modelling. In *CADE*, volume 3632 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2005.

[20] K. Verma, H. Seidl, and T. Schwentick. On the complexity of equational horn clauses. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3328 of *Lecture Notes in Computer Science*, pages 337–352. Springer-Verlag, 2005.